10/607,567 PTO-892

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 17/30

A2

(11) International Publication Number: WO 99/28835

(43) International Publication Date: 10 June 1999 (10.06.99)

(21) International Application Number: PCT/IB98/01909

(22) International Filing Date: 30 November 1998 (30.11.98)

(30) Priority Data: 08/982,972 2 December 1997 (02.12.97) US

(81) Designated States: JP, KR, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published

Without international search report and to be republished upon receipt of that report.

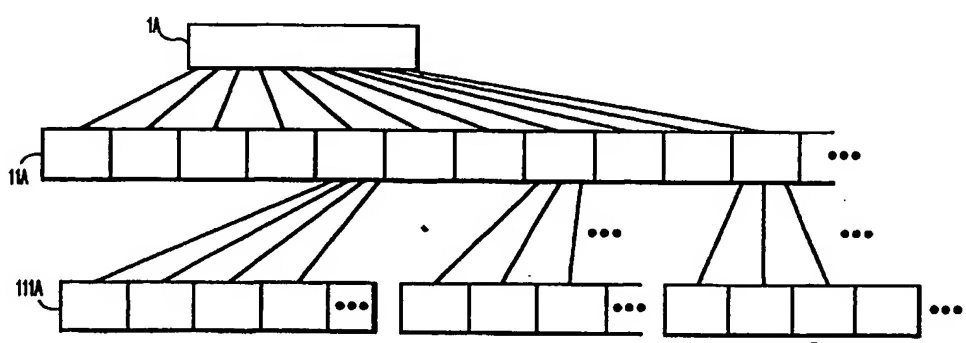
(71) Applicant: KONINKLIJKE PHILIPS ELECTRONICS N.V. [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).

(71) Applicant *(for SE only)*: PHILIPS AB [SE/SE]; Kottbygatan 7, Kista, Kista, S-164 85 Stockholm (SE).

(72) Inventors: ELENBAAS, Herman; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). DIMITROVA, Nevenka; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).

(74) Agent: FAESSEN, Louis, M., H.; Internationaal Octrooibureau B.V., P.O. Box 220, NL-5600 AE Eindhoven (NL).

(54) Title: APPARATUS AND METHOD FOR OPTIMIZING KEYFRAME AND BLOB RETRIEVAL AND STORAGE



(57) Abstract

Keyframes extracted from a video or blobs (binary large objects) extracted from a multimedia or hypermedia document, are stored so as to optimize retrieval from a relatively slow memory device. Retrieval and storage may be based on significant or representative images in the video or on user preferences on information, including text, audio, video and hyperlinks to other documents, from a multimedia document. Elimination of redundant information is also possible.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

Apparatus and method for optimizing keyframe and blob retrieval and storage.

Keyframes extracted from a video or blobs (binary large objects) extracted from a multimedia or hypermedia document or web pages, are stored so as to optimize retrieval from a relatively slow memory device.

Background of the Invention

In a video indexing process, keyframes that visually describe a video may be extracted by video cut detection and keyframe filtering such as described in pending patent applications "Significant Scene Detection and Frame Filtering for a Visual Indexing System", U.S. Serial No. 08/867,140 and "Video Indexing System", U.S. Serial No. 08/867,145, having amongst their inventors, the inventors of the present invention, to create an index. In video cut detection and keyframe filtering, keyframes are selected from a large number of possible frames (30 frames per second of video, typically). Even after the keyframe filtering process, the number of keyframes is considerable, approximately 250 keyframes per video tape. Typically then, the size of an index is approximately 1 MB, if the keyframes are scaled down to 160 x 120 resolution and compressed into JPEG format. Without scaling and compression, the size of the index could be 50 MB or more. At this size, retrieval of keyframes could take considerable amount of time, especially if the retrieval is performed over slow channels such as high latency networks (e.g., Internet, Intranet, etc.) or linear tape mediums such as VHS tape.

Similarly, for web sites, web pages or multimedia or hypermedia documents including blobs are presented. A multimedia document or web page containing video (or images) can require a large amount of memory which may be on the order of tens of megabytes. Time required to download such a multimedia document or software may be considerable with a typical 28.8 kb/sec modem.

A website may include a large number of possible web pages, multimedia documents and links which may be unwieldy for a user to navigate. Each multimedia document or web page may include blobs. The blobs may include audio, video, text, hypertext links or links to other documents. A website retrieval of pages or multimedia documents and their respective blobs, especially those a user has an interest in, may take a considerable

amount of time as blobs typically are stored in temporal or static hierarchies. A multimedia document may be created which provides a user with web pages having audio, video, text and links based on user preferences or other prespecified criteria.

Optimizing retrieval of the keyframes or blobs in a user friendly manner, the index or multimedia document is created using a hierarchical structure representation. Although temporal hierarchies have been used in the literature, such as Ueda, Hirotada and Takafumi Miyatake. "Automatic Scene Separation and Tree Structure GUI for Video Editing" The Fourth ACM International Multimedia Conference Multimedia (November 18-22, 1996): 405-406, as a conceptual representation of keyframes, the present invention creates a linear index structure or linear multimedia document structure out of the temporal hierarchy allowing for optimized retrieval. Currently, storage in databases is typically not optimized for retrieval, but instead, optimized for transaction processing. For example, database systems are optimized for transaction processing such as editing data (i.e., inserting, updating and deleting data) in a database of the system. Query optimization is available also; however, benchmarks of database systems concentrate on changing data as fast as possible with parallel requests.

In databases, order of retrieval is not known in advance since database management systems typically have no knowledge of stored data content or what query will be requested.

In a Digital Compact Cassette (DCC) format, an index system describes which tracks are on a specific tape; however, priority between different tracks does not exist; therefore, optimization of retrieval of the content is not possible.

For a Web page or another similar type multimedia document, information is provided to a user based on a format prespecified by the provider, not on a user-stored preference.

Summary of the Present Invention

A system is desired which optimizes access of an index or multimedia documents. The present invention groups keyframes in nodes and blobs and structures and stores them in a hierarchical manner. The hierarchy includes nodes which are parent or child nodes and blobs based on prespecified user preferences. The number of keyframes (images) in a node and the number of child nodes under a parent node are arbitrary.

Brief Description of the Drawings

Figure 1 illustrates a sample visual index hierarchy;

Figures 2A-2B illustrates a visual hierarchy for the present invention;

Figure 3 illustrates a sample header file;

Figures 4A-4B illustrate hierarchies with group headers;

Figure 5 illustrates a linear representation of the hierarchy;

Figures 6A-6E illustrate detailed representations of the hierarchy; and

Figures 7A-7B are systems of the present invention.

Description of Preferred Embodiments

The present invention includes nodes of keyframes or blobs and links in a hierarchy as illustrated in Figure 1. Although keyframes are referred to in the description, the description is also applicable to blobs.

In the present invention, as shown in Figure 2A, six keyframes are in a node and a maximum of thirty-six child nodes (six child keyframes per parent keyframe) are under a parent node. Clearly, one skilled in the art could modify the number of nodes or number of child nodes under a parent node.

For reference, the top level of nodes (in this example, one node having six keyframes) is Level A, with keyframes labeled 1, 2...x. The second level of nodes is Level B, and includes six nodes. The keyframes are labeled 11, 12, 13...16, 21, 22, 23...26, 31, 32, 33...36,...; and the keyframes on the third level, Level C, are labeled 111, 112...116, 121, 122...126,... The keyframes are numbered, for easy reference and illustration only, to indicate their level and order in the level. The various levels of the hierarchy correspond to the level of detail shown with respect to the underlying video, in this example, with decreasing representation of the video as a whole. For example, those keyframes on Level A are the six most representative frames of the video while those keyframes on Level B are the next most representative and on Level C, the next representative.

An example of the hierarchy presented in Figure 2A for a video which is six hours long and partitioned into x time parts. In this example, the top nodes on Level A (only one node is shown), each have six parent keyframes that represent the entire video and each parent keyframe has six child keyframes. Each of the six keyframes may correspond to one hour of the entire video, thus partioning the video in equal blocks of hours or may correspond to periods of time based on video program structure.

The keyframes on Level B provide more details about the portion of the video tape represented by the parent keyframe. Specifically, keyframes 11, 12, 13...16 under keyframe 1 provide more detail about the first block of time which keyframe 1 represents.

Every keyframe represents a portion of video. For this example, six keyframes are selected to represent the entire video as parent keyframes (Level A), thirty-six keyframes are selected to represent the entire video as child keyframes (Level B) and two hundred and sixteen keyframes are selected to represent the entire video as grandchild keyframes (Level C). Each next level of nodes contains keyframes which are representative of each portion of video of the parent node.

For example, node 1 has all the details of the first portion of the video as represented by six keyframes (1-6). On the next level, keyframe 1, for example, is further detailed by six child keyframes 11-16. On the next level, keyframe 11, for example, is further detailed by six grandchild keyframes 111-116.

The hierarchy created does not necessarily represent a balanced tree. Additionally, the keyframe 1 may be the same as keyframe 11 and keyframe 111.

The temporal hierarchy can be stored on a memory device such as a disk or tape using many different structures. In the present invention, the hierarchy is "flattened" for storage in a computer-readable medium by describing the structure in a header file and by grouping the keyframes in independent nodes. For a file, in this example, the filenames of the keyframes represent associated time information in one thirtieth of seconds.

Additional more descriptive information from an associated visual index may also be included in the header file, as is done in the present invention. Information in this file is presented in attribute-value pairs at three levels: tape, node and frame. The attribute-value pairs of the present structure gives freedom for inserting new attributes, for example, levels for classification of the tapes or objects within a frame.

Similarly, the present invention may be used for providing and/or retrieving multimedia documents or hypermedia documents such as a web page. A user may have specific interests allowing a user profile or user preference information to be created by a server who may then package information dynamically. For example, as shown in Figure 2B, a document (Document) 1A may contain audio, video (images), text and/or links to other documents (Doc1, Doc2, Doc3, etc.) 11A. A user may only have interest in information contained in some of the audio, video, text or these other documents, for example, Doc2 and Doc3, and not others, Doc1. Each further document, Doc1-Doc3, may include text, audio, and/or video and further links to still further multimedia documents 111A.

The hierarchy created does not necessarily represent how a user would wish to retrieve the information (audio, video, text, and/or links) or have any relation to a user's preference. An analysis can be performed on the information based on a prespecified user

profile and the information can be reordered into a temporal hierarchy by "flattening" the reordered hierarchy into a user file which is embodied on a computer-readable medium.

Figure 3 illustrates a sample header file. A header may include such information as video tape ID, title of the video, category of the video, recording date, index date, tape length, version of the visual index, resolution of the images, number of levels, number of child nodes, and number of key frames in the visual index. This information codes frame numbers and information and based on the coded frame numbers and information, can calculate from which position on the storage, i.e., video tape, CD, a VCR should be positioned. It may be desired to limit the information stored in the header file to prevent data corruption and to reduce storage. Additionally, the header file could be stored in several places on the storage medium to prevent data corruption.

In this example, a visual index contains a header file (video header) 410 or 416 and the keyframes or keyframe images 412 & 414 or 418 and 420. The visual index of, in this example, 216 keyframe images has a header file of 4 KB while the keyframe images take 844 KB. Although in the present example, one header file is used which may be specific or general to the video, level or group headers (422 and 424 or 426 and 428) could be added to describe specific levels of nodes as shown in Figures 4A and 4B, as could other types of headers.

Figure 4A illustrates a hierarchical level wise keyframe clustering while Figure 4B illustrates a parent-child wise clustering of keyframes for storage.

Figure 5 illustrates a visual index structure which flattens and linearly represents the hierarchy. In an archiving process, this structure is created on a temporary device such as a disk or other computer-readable medium and written in its entirety to a linear medium, such as a tape or over a network. In the present invention, the header file is the first file to allow easy access to information saved in the visual index. Ordering of keyframe image node files is done depending on the rendering of the hierarchical temporal structure.

Depending on the user interface, the nodes of keyframes are ordered in a selected structure and saved. Several different structures are possible, as shown in Figures 6A-E. Specifically, Figure 6A illustrates a hierarchical top-down ordering, Figure 6B illustrates a left-right ordering and Figure 6C illustrates a level ordering. Figure 6D illustrates a level ordering which eliminates redundant storing of same frames. Specifically, as previously mentioned, keyframe 1, 11, and 111 may represent the same image and thus, storage of all three is redundant. Thus, only keyframe 1, for example, is stored.

Figure 6E illustrates an ordering for a multimedia document which eliminates links to other documents, text, audio, or video, in which a user has indicated disinterest, to provide a user file. Figure 6E provides an example ordering for the example described in Figure 2B.

In all orderings, a node header, if used, may include such information as ID, number of key frames for the specific level, and for each key frame, ID, annotation, position, number of child nodes and frame signature.

Node images may also be included. For each keyframe, information such as ID and image data may be included.

To retrieve the saved keyframes, the header file is read first, then the Level A first keyframes or blobs are read and stored on a temporary device such as a disk or other computer-readable medium. To optimize retrieval of the visual index or multimedia document, the visual index or multimedia document is restored in different segments. After each segment is read, the information can be displayed to the user. Thus, a user does not have to wait for the entire visual index or multimedia document to load to look at levels or areas of interest already loaded. A user may see the most representative keyframes or blobs of interest and progress toward more detail as the visual index or web page, respectively is being loaded. At the moment the keyframe image node or blob that for the user interface is read, it is sent to a memory from where the images, etc. may be displayed to the user. Finally, the other keyframe images or blobs are loaded in a prespecified order.

Figures 7A and 7B illustrate example systems of the present invention. Specifically, in Figure 7A, a storage 702 has a selected number of most representative keyframes as provided by a video indexing system or other automatic or manual means. The storage 702 provides the selected keyframes to a first processor 704 which orders the keyframes into a selected number of levels, each level including a predetermined number of the most representative keyframes and each subsequent level including a multiple number of keyframes of the previous level. A second processor 705, which may be a separate second processor 705 or a part of the first processor 704, creates at least one header file based on information about the most representative keyframes of the video.

The header file and keyframes are embodied in an index file in a memory 706 which may be a separate memory or part of the storage 702. A unit 708 which may be a separate device such as a computer, VCR, or television and may have a user-interface, then retrieves the index file and presents the keyframes for each level, as each level is retrieved.

Similarly, the example system in Figure 7B has a storage 710 which may be, for example, present in a server. A first processor 712 would order blobs into a selected number of levels. Each level would include at least one of text, video, audio and links to other multimedia or hypertext documents. Each subsequent level would include at least one of text, video, audio and further links for each of the other multimedia or hypertext documents.

A second processor 713, which may be a separate processor or part of the first processor 712, would organize blobs into a user file based on user preference information. The second processor 713 would be able to analyze a blob or link against a database or based on embedded information, to determine if the blob or link falls within a user's prespecified area of interest. The second processor then organizes blobs and links based on this analysis to present those blobs and links at the top of a user's prespecified areas of interest first, such as was shown in Figure 6F.

A memory 714, which may be a separate storage or part of the storage 710 would store the organized blobs and links embodied in the user file. A unit 716, such as a computer, would retrieve the user file and present the blobs and links, as each is retrieved.

As can now be readily appreciated, the invention allows storage of keyframes or blobs so as to optimize retrieval from a relatively slow memory device. The invention may be included in any of the subsystems or may be a separate subsystem. One skilled in the art may easily use differing numbers of nodes, keyframes, blobs, headers, node headers and node images. Additional modifications may easily be made by one skilled in the art.

The present invention may also be expanded to include video clips, audio (sound, speech, music, etc.), colors or video characteristics, and/or annotation, text or data (manually or automatically added) in conjunction with or presented separately with the keyframes.

Further, a master index could be stored for a collection of video tapes, files, etc. allowing a user to view the master index which may include information as to where specific programs, segments, etc. are stored.

The keyframes could also be analyzed and consequently, reorganized according to prespecified criteria such as user preferences or various clustering methods, such as shown in Figures 4A and 4B. This would permit storage such that those keyframes which are indicated as having a higher priority are stored first in the data structure of the index file to permit earlier retrieval.

It will thus be seen that the objects set forth above among those made apparent from the preceding description, are efficiently attained and, since certain changes may be

made in the above constructions without departing from the spirit and scope of the invention, it is intended that all matter contained in the above description or shown in the accompanying drawings shall be interpreted as illustrative and not limiting sense.

It is also to be understood that the following claims are intended to cover all of the generic and specific features of the invention herein described and all statements of the scope of the invention which, as a matter of language, might be said to fall therebetween.

CLAIMS:

1. A method for storing keyframes of a video in computer-readable medium to optimize retrieval, said method comprising the steps of:

retrieving, from the computer-readable medium, a selected number of most representative keyframes and ordering keyframes into a selected number of levels, each level including a predetermined number of the most representative keyframes and each subsequent level including more keyframes representative of the video; and

storing the keyframes by level in a computer-readable medium.

2. A method for storing keyframes as recited in Claim 1, further comprising the steps of:

creating at least one header file based on information about the most representative keyframes of the video; and

storing the at least one header file in the computer-readable medium.

3. A method for presenting keyframes of a video, said method comprising the steps of:

retrieving a selected number of most representative keyframes of the video and ordering retrieved keyframes into a selected number of levels, each level including a predetermined number of the most representative keyframes and each subsequent level including a predetermined number of keyframes decreasingly representative of the video;

storing the keyframes by level embodied in an index file in an computer-readable medium;

retrieving the index file; and

presenting the keyframes as each level is retrieved.

4. A system for storing keyframes of a video in memory, said system comprising: storage having a selected number of most representative keyframes of the video;

a first processor ordering keyframes into a selected number of levels, each level including a predetermined number of the most representative keyframes and each subsequent level including a multiple number of keyframes of the previous level;

a second processor creating at least one header file based on information about the most representative keyframes of the video;

memory storing the header file and the keyframes by level and embodied in an index file; and

a unit for retrieving the index file and presenting the keyframes for each level, as each level is retrieved.

5. A method for storing blobs of a multimedia document for optimizing or personalizing retrieval, in a computer-readable medium, said method comprising the steps of:

creating a user preference file based on prespecified information;

retrieving blobs for a specific multimedia document and ordering blobs into a selected number of levels, each level including at least one of blobs of the specific multimedia document and links to other multimedia documents and each subsequent level including blobs of the other multimedia documents; and

storing specific blobs and links based on the user preference file embodied into a user file for retrieval by a user.

6. A method for presenting blobs and links of a multimedia document, said method comprising the steps of:

retrieving selected at least one of blobs and links and ordering the selected blobs and links into a selected number of levels, each level including at least one of audio, video, text and links to other multimedia documents;

creating a user file based on user preference information, said user file embodying the blobs and links;

retrieving the user file; and

presenting the blobs and links as each is retrieved.

7. A system for storing blobs of a multimedia document, said system comprising:

storage having blobs for each multimedia document;

a first processor ordering blobs for a selected multimedia document into a selected number of levels, each level including at least one of text, video, audio and links to

other multimedia documents, each subsequent level including at least one of text, video, audio and links to still further multimedia documents for each of the other multimedia documents;

a second processor to organize blobs and links based on user preference information;

memory storing the organized blobs embodied in a user file; and

a unit for retrieving the user file and presenting the blobs and links, as each is retrieved.

8. A computer-readable medium embodying a data structure comprising:

keyframes of a first level;

keyframes of a second level; and

keyframes of further levels in a prespecified level order.

9. A computer-readable medium embodying a data structure comprising:

t least one of blobs of a first multimedia document and links to other multimedia documents in an order based on a user preference file;

t least one of blobs of the other multimedia documents and links to still further multimedia documents in an order based on the user preference file; and

further blobs and links of the further multimedia documents in an order based on the user preference file.

10. A computer-readable medium embodying a data structure comprising:

keyframes of a first priority;

keyframes of a second priority; and

keyframes of further priorities in a prespecified priority order.

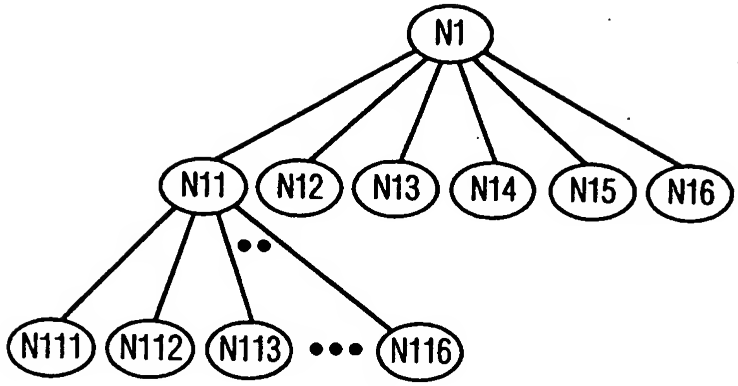


FIG. 1

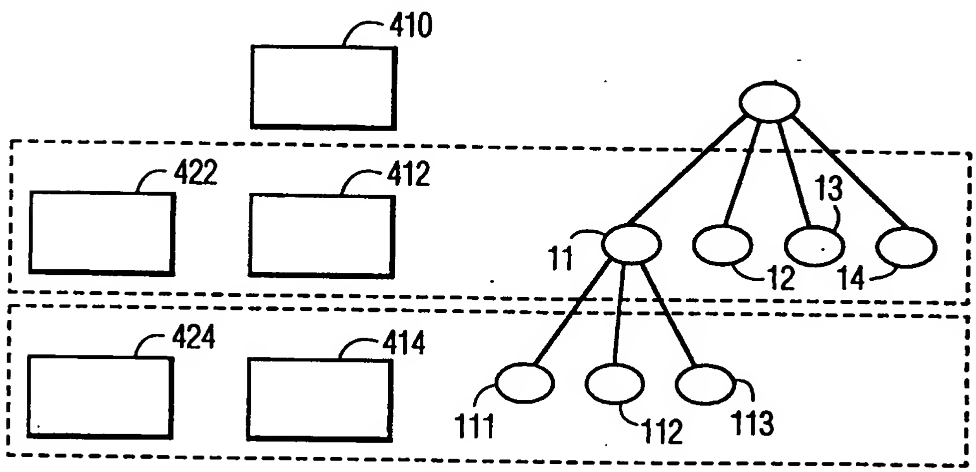


FIG. 4A

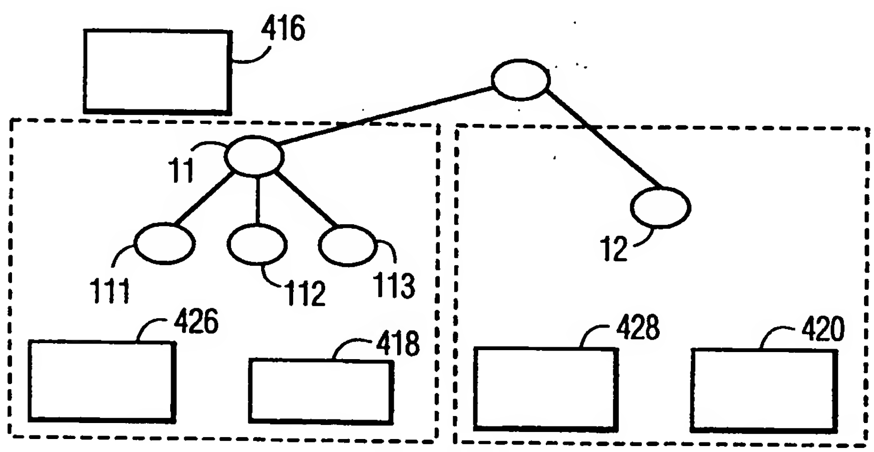


FIG. 4B

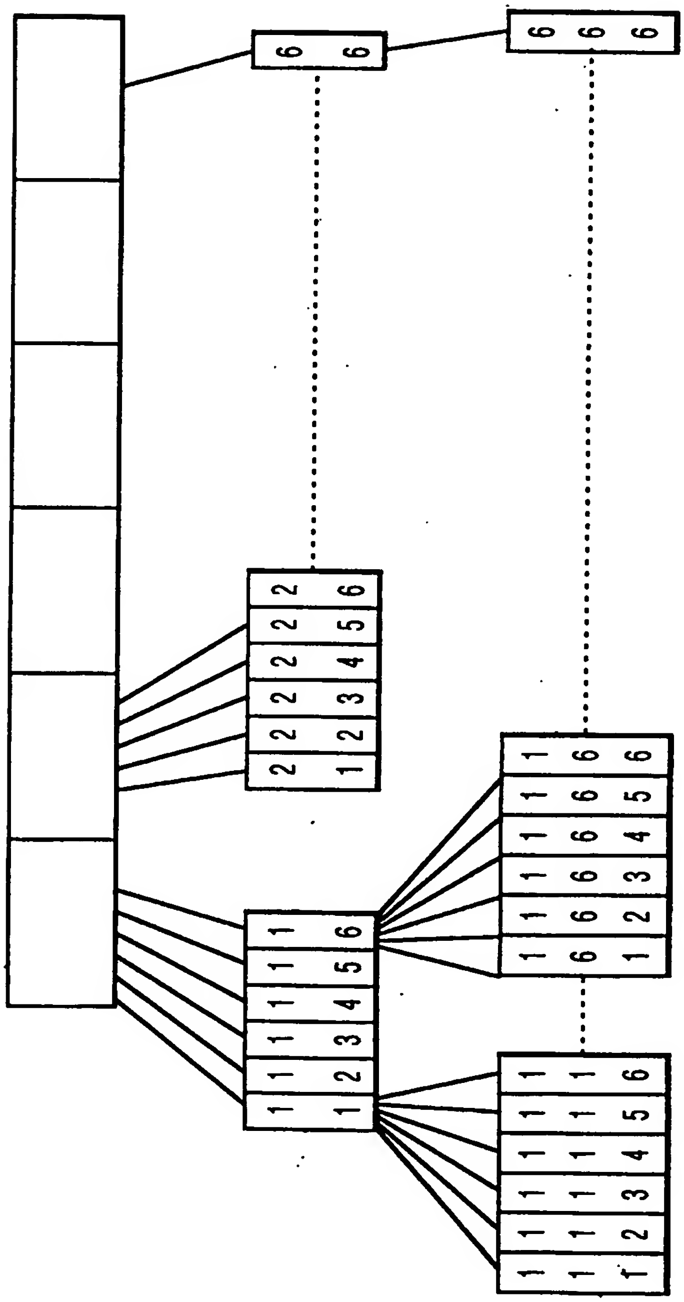


FIG. 2A

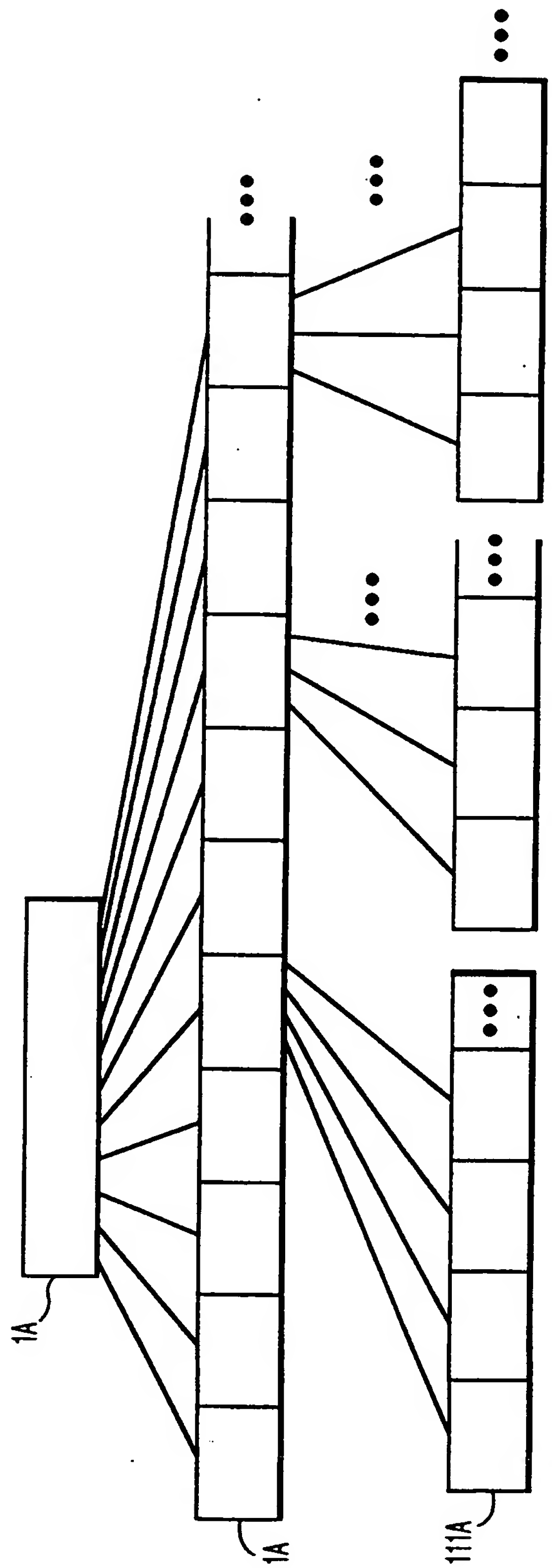


FIG. 2B

= 1.0
= 0001
= f0000000.jpg|f0010928.jpg|f0019830.jpg|f0025663.jpg|f0033606.jpg|f0042474.jpg

= 0011
= f0003628.jpg|f0004461.jpg|f0007009.jpg|f0008036.jpg|f0009586.jpg

= 0012
= f0012266.jpg|f0013543.jpg|f0015570.jpg|f0017307.jpg|f0018907.jpg

= 0013
= f0020580.jpg|f0022215.jpg|f0023207.jpg|f0023695.jpg|f0024006.jpg

= 0015
= f0034906.jpg|f0036494.jpg|f0039387.jpg|f0040506.jpg|f0041608.jpg

= 0016
= f0043543.jpg|f0044384.jpg|f0045894.jpg|f0047984.jpg|f0049012.jpg

= 0111
= f0001386.jpg|f0002000.jpg|f0002752.jpg|f0003496.jpg|f0003497.jpg

= 0112
= f0003753.jpg|f0003918.jpg|f0003923.jpg|f0004185.jpg|f0004355.jpg

...
.....
= 0165
= f0048121.jpg|f0048296.jpg|f0048357.jpg|f0048432.jpg|f0048698.jpg

= 0166
= f0049178.jpg|f0049264.jpg|f0049434.jpg|f0049555.jpg|f0050015.jpg

FIG. 3

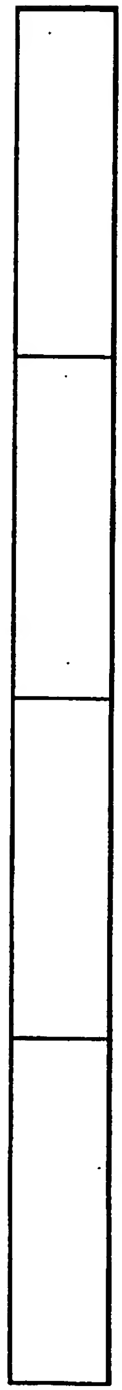


FIG. 5

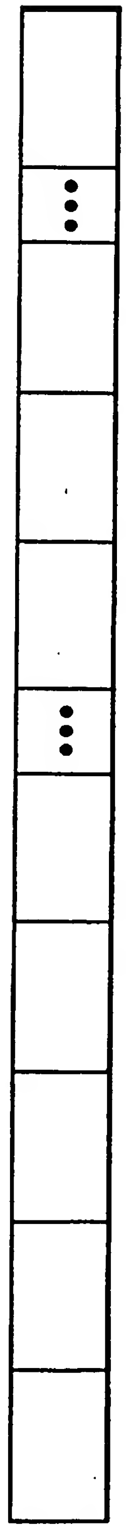


FIG. 6A

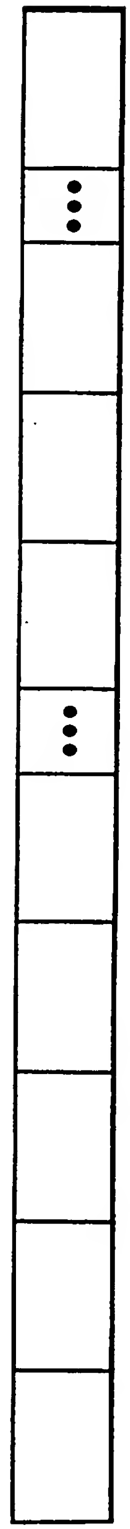


FIG. 6B

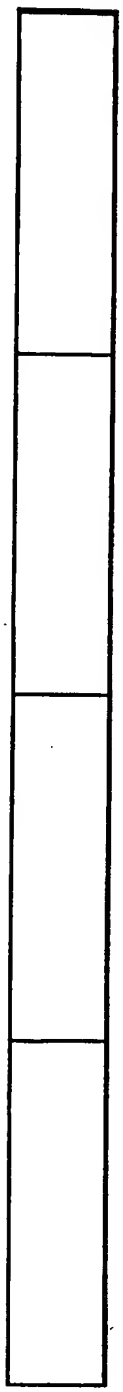


FIG. 6C

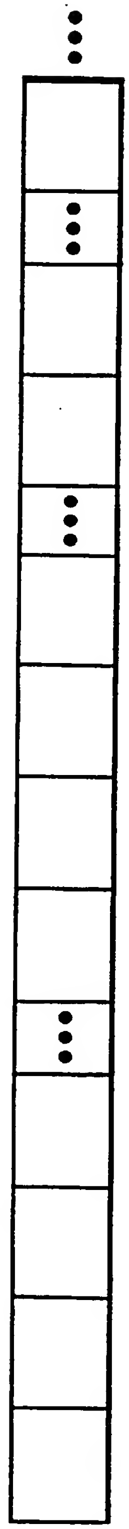


FIG. 6D

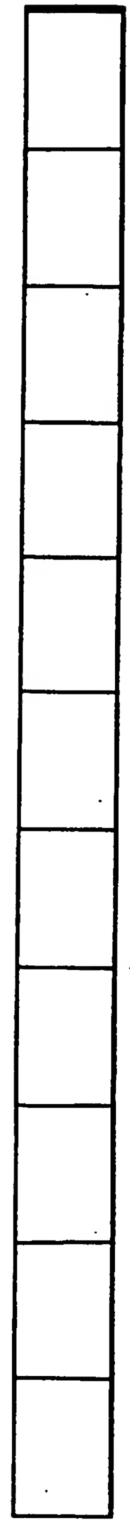


FIG. 6E

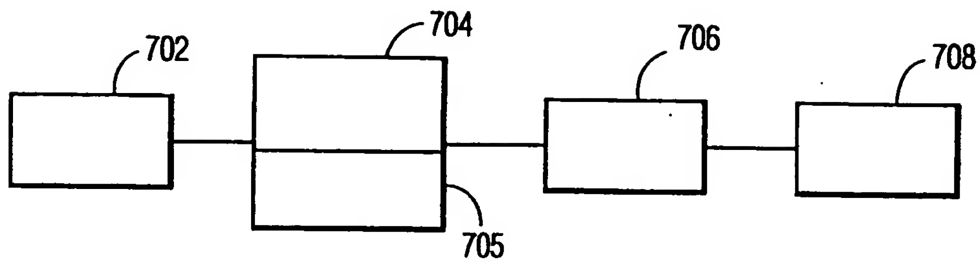


FIG. 7A

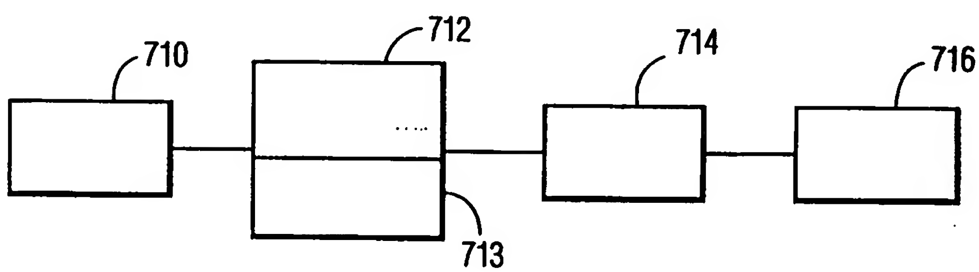


FIG. 7B